Claim Listing:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (original) A method, comprising:

(a) maintaining a device identifier and a private key in a programmable logic device, the device identifier and the private key being non-volatile such that if power to the programmable logic device is lost the device identifier and private key remain stored in the programmable logic device;

(b) receiving a first encrypted key onto the programmable logic device, and using the device identifier and the private key to decrypt the first encrypted key thereby generating a first key;

(c) receiving onto the programmable logic device a bitstream comprising first encrypted configuration data encrypted with the first key;

(d) using the first key to decrypt the first encrypted configuration data on the programmable logic device thereby generating first configuration data; and

(e) configuring a first portion of the programmable logic device using the first configuration data.

2. (original) The method of Claim 1, wherein neither the device identifier nor the private key are rewritable.

3. (original) The method of Claim 1, wherein the bitstream further comprises a first key number associated with the first encrypted configuration data, the first key being stored on the programmable logic device in association with the first key number, the programmable logic device in step (d) using the first key number in the bitstream to identify the first key as the key that will be used in step (d) to decrypt the first encrypted configuration data.

4. (original) The method of Claim 1, wherein the device identifier and the private key are stored on the programmable logic device in one of the group consisting of: an antifuse-based storage element, a fuse-based storage element, a laser-programmed storage element, an EPROM storage element, and a flash-based storage element.

5. (original) The method of Claim 1, further comprising:

after the first key is generated in step (b), storing the first key in non-volatile memory on the programmable logic device.

6. (original) The method of Claim 1, wherein the first encrypted configuration data is decrypted in step (d) on the programmable logic device by a hardware decryptor.

7. (original) The method of Claim 1 wherein

step (b) further comprises receiving a second encrypted key onto the programmable logic device and using the device identifier and the private key to decrypt the second encrypted key, thereby generating a second key;

step (c) further comprises receiving onto the programmable logic device a bitstream comprising second encrypted configuration data encrypted with the second key;

step (d) further comprises using the second key to decrypt the second encrypted configuration data on the programmable logic device, thereby generating second configuration data; and

step (e) further comprises configuring a second portion of the programmable logic device using the second configuration data.

8. (original) The method of Claim 7, wherein the bitstream further comprises a first key number associated with the first encrypted configuration data, and wherein the bitstream further comprises a second key number associated with the second encrypted configuration data, the first key being stored on the programmable logic device in association with the first key number, the second key being stored on the programmable logic device in association with the second key number, the programmable logic device in (d) using the first key number in the bitstream to identify the first key as the key that will be used in (d) to decrypt the first encrypted configuration data, the programmable logic device in (d) using the second key number in the bitstream to identify the second key as the key that will be used in (d) to decrypt the second configuration data.

9. (original) The method of Claim 7 further comprising:

after the first key and the second key are generated in step (b), storing the first key and the second key in non-volatile memory on the programmable logic device.

10. (original) The method of Claim 7, wherein the first portion of the programmable logic device is configured in (e) to realize a first IP module, and wherein the second portion of the programmable logic device is configured in (e) to realize a second IP module.

11. (original) The method of Claim 1, wherein the programmable logic device is an SRAM-based PLD.

12. (original)The method of Claim 10, wherein the non-volatile memory in the programmable logic device is flash-based.

13. (original) The method of Claim 10, wherein the non-volatile memory in the programmable logic device is one-time programmable.

14. (original) The method of Claim 10, wherein the non-volatile memory in the programmable logic device is antifuse-based.

15. (original) The method of Claim 10, wherein the non-volatile memory in the programmable logic device is fuse-based.

16. (original) The method of Claim 1, wherein the device identifier and the private key are rewritable at one time, but as of the time step (a) occurs are no longer rewritable.

17. (previously presented) The method of Claim 1, further comprising:

receiving on a license manager the device identifier maintained on the programmable logic device;

receiving on the license manager a first authorization code; and

determining whether the first authorization code has a predetermined relationship with respect to the device identifier, wherein if the first authorization code is determined to have the predetermined relationship then the license manager sends the first encrypted key to the programmable logic device such that it is received in step (b), and wherein if the first authorization code is determined not to have the predetermined relationship then the license manager does not send the first encrypted key to the programmable logic device in step (b).

18. (original) The method of Claim 17, wherein the first authorization code has the predetermined relationship with respect to the device identifier if the first authorization code contains the device identifier in an encrypted form.

19. (previously presented) A method comprising:

receiving onto a programmable logic device an encrypted

first key;

on the programmable logic device decrypting the encrypted first key to generate a first key and storing the first key on the programmable logic device;

receiving onto the programmable logic device a configuration bitstream having a first portion and a second portion;

on the programmable logic device decrypting the first portion of the configuration bitstream using the first key; and

configuring the programmable logic device with the decrypted first portion of the configuration bitstream thereby realizing a first IP module.

20. (original) A programmable logic device that receives an encrypted configuration bitstream, the programmable logic device comprising:

non-volatile storage that stores a first key;

a decryptor that decrypts a first part of the encrypted configuration bitstream using the first key and thereby generates first configuration data; and

first configurable logic elements being configured by the first configuration data.

21. (original) The programmable logic device of Claim 20, wherein the first part of the encrypted configuration bitstream is identified by a first key number in the bitstream, the first key number identifying the first key in the non-volatile storage.

22. (original) A method, comprising:

receiving on a development system a device identifier from a programmable logic device;

receiving on the development system an authorization code;

verifying on the development system that the

authorization code and the device identifier have a predetermined relationship, wherein if the authorization code and the device identifier have the predetermined relationship then encrypting a key using the device identifier and sending the encrypted key from the development system to the programmable logic device, but wherein if the authorization code and the device identifier do not have the predetermined relationship then the encrypted key is not sent from the development system to the programmable logic device; and

the development system using the key to encrypt a portion of a configuration data bitstream, the development system outputting the configuration data bitstream including the encrypted portion.

23. (original) The method of Claim 22, wherein the key has a key number, and wherein the development system adds the key number to the configuration data bitstream such that the key number is associated with the encrypted portion of the configuration data bitstream, the configuration data bitstream output from the development system including the encrypted portion and the key number.

24. (original) The method of Claim 22, wherein the development system comprises a capture/design tool and a license manager, the method further comprising:

if the authorization code and the device identifier are verified as having the predetermined relationship then the license manager allows use of IP module design information by the capture/design tool, whereas if the authorization code and the device identifier are not verified as having the predetermined relationship then the license manager does not allow use of the IP module design information by the capture/design tool.

25. (original) The method of Claim 22, wherein the portion of the configuration data bitstream is configuration data for an IP module, the development system comprising a capture/design tool, the capture/design tool being usable to view a net external to the IP module, the capture/design tool not being usable to view a net internal to the IP module.

26. (original) A development system, comprising:

a capture/design tool; and

means for verifying that an authorization code has a predetermined relationship with respect to a device identifier read from a programmable logic device, and if the authorization code is verified then the means also encrypting a key and sending the encrypted key to the programmable logic device, if the authorization code is verified then the means also uses the key to encrypt a portion of a configuration data bitstream output by the capture/design tool, the configuration data bitstream including the encrypted portion being sent to the programmable logic device.

27. (original) The development system of Claim 26, wherein the encrypted portion of the bitstream contains configuration data for an IP module, the capture/design tool being usable to view a net external to the IP module, the capture/design tool being unusable to view a net internal to the IP module.

28. (original) The development system of Claim 26, wherein the key has a key number, the means inserting the key number into the configuration data bitstream sent to the programmable logic device, the key number in the configuration data bitstream being associated with the encrypted portion of the configuration data bitstream.

29. (previously presented) The method of Claim 19, further comprising:

receiving onto the programmable logic device an encrypted second key;

on the programmable logic device decrypting the encrypted second key to generate a second key and storing the second key on the programmable logic device;

on the programmable logic device decrypting the second portion of the configuration bitstream using the second key; and configuring the programmable logic device with the decrypted second portion of the configuration bitstream thereby realizing a second IP module.